Logan DELPORTE et Alan GARCIA CALZADA | DUT Informatique 2ème année TD B

RAPPORT DE PROJET JAVA

Sommaire

Description du projet

Le but du projet est de réaliser un jeu inspiré de Reigns avec des éléments de Tropico en java. Le jeu reprend les mécaniques de Reigns tout en reprenant l'univers et l'ambiance de Tropico.

Il s'agit d'un jeu de gestion se déroulant sous la forme d'une suite de questions influant sur des statistiques. Le but étant de tenir le plus longtemps possible sans que le joueur fasse faillite ou que son peuple se révolte.

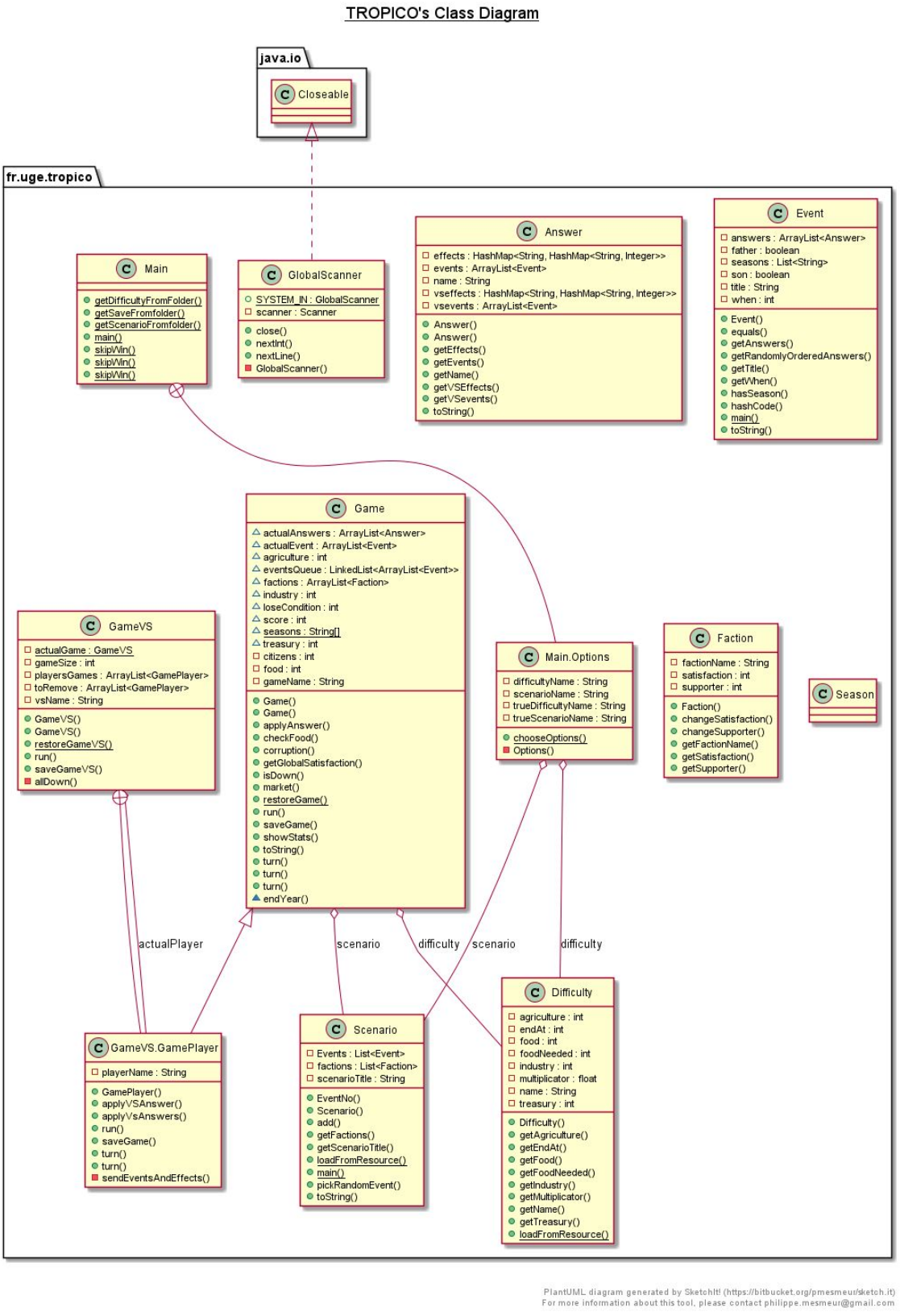
Le jeu est jouable en solo ainsi qu'en multijoueur.

Le jeu lance un scénario ainsi qu'un niveau de difficulté à partir d'un fichier json comprenant tous les paramètres nécessaires au jeu.
De par sa structure, il est aisément moddable et quiconque étant capable de modifier le fichier json, afin de modifier tous les paramètres du jeu à sa guise (faction, multiplicateur d'effets, quantité initiale de nourriture/argent, niveau d'industrialisation/agriculture…).

Le jeu dispose d'un système de sauvegarde. Pour éviter tout cas de triche, nous avons préféré un système de sauvegarde automatique à chaque événement plutôt qu'un système manuel. Il est possible de charger le fichier de sauvegarde en lançant une partie.

Pour faciliter la création de scénarios, nous avons créé un programme annexe (en python) permettant de rentrer les différents paramètres du scénario (factions, events…) et d'éditer automatiquement le fichier json correspondant au scénario demandé.

Présentation de l'architecture



(Pour un diagramme plus complet veuillez faire regarder le fichier TropicoUMLPlus.png)

L'architecture du programme est constituée de différentes classes concernant des éléments du jeu et communiquant entre eux si besoin (par exemple, la classe Event communique avec la classe Answer pour accéder aux fonctions manipulant les réponses de l'événement).

La classe Main est appelée au lancement du programme, une fois tous les paramètres rentrés, celle-ci fait appel à la fonction run() de la classe Game permettant le déroulement du jeu. La classe Game est celle où toutes les autres classes "gravitent" autour.

Répartition des tâches

Alan GARCIA CALZADA :

- Lecture et interprétation de fichiers json
- Structure globale (événements, réponses, scénario, difficulté)
- Création du menu (classe Main)
- Système de sauvegarde
- Ajout du mode multijoueur

Logan DELPORTE :

- Déroulement de la partie (classe Game)
- fonctionnement des factions
- Création du scénario
- Réalisation de la javadoc
- Rédaction du rapport et du guide utilisateur

Difficultés rencontrées

Le projet s'est assez bien déroulé, les quelques problèmes que nous avons rencontré étaient des bugs qui ont été plutôt facilement réglés. Nous aurions voulu faire une interface graphique mais nous avons trouvé plus judicieux de régler les bugs et faire en sorte que le jeu soit “modifiable” le plus possible.

De plus, l'aspect de la console se porte bien avec ce type de jeu (un peu comme un chat).

Possibles ajouts

Les potentiels ajouts que nous envisageons seraient tout d'abord de rendre le code un peu plus "propre".

Ensuite, il serait possible d'implémenter une interface graphique pour le jeu.

Pour élargir les possibilités de modding, nous voudrions permettre au joueur de changer le nombre et le nom des saisons (dans le cas où le joueur souhaite créer un scénario se passant dans une galaxie lointaine, très lointaine... **mélodie très familière d'une saga très connue commence à se jouer**) et aussi pouvoir créer des factions du type "Loyaliste".

Pour un maximum d'ambition nous pouvons éventuellement penser à un mode multijoueur en ligne.

Conclusion

Ce projet nous a permis de nous améliorer en programmation Java dans un cadre passionnant (création d'un jeu) et nous avons également pu améliorer notre efficacité en travail d'équipe.

Tout ça pour la réalisation d'un jeu qui nous satisfait.